New Side Channel Attack Countermeasure Based on Minimal Hamming Weight Distribution (Mathematical Foundation of Algorithms and Computer Science)

AUTHOR(S):

Suppakitpaisarn, Vorapong; Edahiro, Masato

数理解析研究所講究録
第1691巻 2010年 174-180

New Side Channel Attack Countermeasure Based on Minimal Hamming Weight Distribution

Vorapong Suppakitpaisarn * †　　Masato Edahiro ‡

平成22年6月9日

概要

To cope with the side channel attack, many works have been proposed. In this paper, we are interested in the method of varying the representation of the scalar. By using this method, we have to increase the hamming weight of the representation. This makes the computation time of the scalar-point multiplication of the elliptic curve cryptosystem increases. In this paper, we improve the hamming weight of the randomized binary expansion, and make the hamming weight fixed for any scalars. This is done by utilizing the fact that the distributions of the minimal hamming weight are the standard distributions for any binary expansions. As a result, our proposed expansion is randomized expansion as in the work by Ha and Moon [1, 2]. It has a fixed hamming weight as in the work by Mamiya and Miyaji [3]. And, it improves the average hamming weight on both papers from 0.50 to 0.43. Our method can be applied to the multi-scalar point multiplication, and the scalar point multiplication on the enlarged digit set.

1 Introduction

Side channel attack [4] is the utilization of the cryptographic environment to break the cryptosystem. These include the computation time [5], the power consumption [6], or the EM wave transmitted from the cryptosystem [7]. This method is shown that it can be used for breaking the scalar point multiplication in the elliptic curve cryptosystem that utilizes the double-and-add method. This is because of the fact that the power consumption using for point additions and point doubles are different. Also, the time used on point additions depends on the hamming weight, the number of non-zero digit on the expansion of the scalar. Many works have been proposed to cope with this problem. These include inserting the dummy point additions [8], making the point additions and point doubles indistinguishable [9], or group some point additions and point doubles into indistinguishable blocks [10]. Each scheme has its own advantages and disadvantages. For instances, the dummy additions are easy to implement, but its efficiency is poor, and it is weak against the fault analysis attack [11]. The indistinguishable operation and the indistinguishable block are strong against most of attacks, but the implementation is hard. It is difficult to apply the idea to the different type of the elliptic curve, or the different scalar representation.

In this paper, we are interested in the method that varying the representation of the scalar. As representing the number using a redundant digit set is intensively used for improving the computation time of the scalar-point multiplication, it is also able to be used for preventing the side channel attack. Since one scalar can be represented by many expansions, randomly selected one representation can make the eavesdropper harder to get any information. The first work on this scheme is done by Ha and Moon [1, 2]. They propose the randomized expansion on digit set $\{0, \pm 1\}$. Although, their works have been proved to be weak against many attacks or implementation environments [12, 13, 14], it is still intensively used. This is because of the fact that the implementation is not very hard, and independent to the type of the curve. However, the binary expansion proposed in [1, 2] has the average hamming weight equals to $\frac{1}{2} = 0.50$. This is much higher than the minimal average hamming weight that is equal to $\frac{1}{3} \approx 0.33$. We note that this number affects the number of point additions needed for the scalar-point multiplication.

Proposed in [3], the fixed-hamming-weight representation is important to prevent the timing attack, especially for the case that the point additions and the point doubles are indistinguishable. They also propose the conversion to make the representation by $\{0, \pm 1\}$ has the fixed hamming weight. That fixed hamming weight is $\frac{1}{2} = 0.50$. Similar to the work by Ha and Moon, this makes the operation slower. Although the minimal average hamming weight is $\frac{1}{3} \approx 0.33$, the worst minimal hamming weight is $\frac{1}{2} = 0.50$. Then, it is hard to reduce the fixed hamming weight.

*Graduate School of Information Science and Technology, The University of Tokyo
†ERATO-SORST Quantum Computation and Information Project, JST
‡System IP Core Research Laboratories, NEC Corporation

In this work, we apply the minimal weight distribution to improve this problem. We prove that the distribution of the minimal weight is always a normal distribution, for any binary expansions. This results is not only limited to a single integer, but also a pair, a triple, or larger number of integers. This fact is obtained by analyzing the Markov chain proposed for automatically finding the average hamming weight on any expansions on our previous work [15, 16]. In this analysis, we are also able to find the expected value and the standard deviation of the distribution. As a results, we know that more than 97.73% of the single scalar has the minimal weight in the digit set $\{0, \pm 1\}$ less than 0.43 when the length of bit string is 160. Then, we propose not to use the scalar which has the weight more than 0.43 as a key, and we randomly increase the weight of the scalar which has the weight less than that number. This idea makes us able to produce the randomized representation which has the fixed hamming weight equals to 0.43. As this work is still on-going, our expansion has not been ensured whether it is strong against the attacks. But, we believe that it is stronger than the work by Ha and Moon.

This paper is organized as follows: On next section, we review the concepts about the scalar-point multiplication. We also describe our method to find the average weight on any digit sets. In Section 3, we describe how our model can be used for proving that the minimal hamming weight distribution is minimal. Also, we explain how to find the standard deviation from the model. In Section 4, we discuss how to apply the minimal weight distribution to the side channel attack countermeasure. And, we conclude the paper, and propose some future works in Section 5.

2 Preliminaries

2.1 Scalar-Point Multiplication

Scalar-point multiplication is the operation to compute

$$S = rP,$$

when r is a natural number, and P is a point in the elliptic curve.

Using the double-and-add method, we can compute the operation efficiently. For example,

$$S = 14P = (1110)_2 P$$

can be computed by

$$S = 2(2(2P + P) + P).$$

This needs 3 point doubles and 2 point additions. The number of point doubles required is constantly equal to $\lfloor \log_2 r \rfloor$. And, the number of point additions required is equal to $W_{bin}(r) - 1$, when $W_{bin}(r)$ is the hamming weight of r in the binary representation.

In some elliptic curves, the point inversions can be done easily. Then, representing the number using the digit set $\{0, \pm 1\}$ can improve the scalar-point multiplication. For example,

$$S = 14P = (100\bar{1}0)_2 P,$$

when $\bar{1} = -1$, can be computed by

$$S = 2(2(2(2P)) - P).$$

This needs 4 point doubles and 1 point addition. Although, the number of point doubles sometimes increases by 1, the number of point additions usually decrease significantly. There is the work [17, 18] presented the minimal expansion on this representation. They call the representation as NAF.

To evaluate the representation E, we use the average weight

$$AW(E) = \lim_{k \to \infty} \sum_{r=0}^{2^k} \frac{W_E(r)}{k 2^k}.$$

For examples,

$$AW(bin) = \frac{1}{2} = 0.50,$$

$$AW(NAF) = \frac{1}{3} \approx 0.33.$$

The generalized version of NAF representing by the digit set $\{0, \pm 1, \ldots, \pm(2^w - 1)\}$ is called $w - NAF$. It is proved that $AW(w - NAF) = \frac{1}{w+2}$ [19].

Next, we consider the multi-scalar point multiplication

$$S = r_1 P_1 + \cdots + r_d P_d,$$

which is used for elliptic curve digital signature algorithm [20]. Instead of computing $r_1 P, \ldots, r_d P_d$ in separate, Shamir's trick can make the operation faster. For example, let $d = 2$,

$$r_1 = 12 = (10\bar{1}00)_2,$$

$$r_2 = 21 = (10101)_2.$$

And, we precompute $D_1 = P_1 + P_2$, $D_2 = P_1 - P_2$. We can compute $S = 12P_1 + 21P_2$ as

$$S = 2(2(2(2D_1) - D_2)) + P_2.$$

This requires 4 point doubles and 2 point additions. Similar to the scalar point multiplication, the number of point doubles required is

$$\max_c(\lfloor \log_2(r_c) \rfloor).$$

And the number of point addition required is equal to $JW_E(r_1,\ldots,r_d)$ - 1. $JW_E(r_1,\ldots,r_d)$ is the joint hamming weight of $r_1,\ldots,r_d$ in the representation E,

$$JW_E(r_1,\ldots,r_d) = ||\{c \in \mathbb{Z} | E_c(r_1,\ldots,r_d) \neq \langle \mathbf{0} \rangle\}||,$$

when $E_c(r_1,\ldots,r_d)$ is $\langle \alpha_t \rangle_{t=1}^d$. α_t is the c^{th} bit when expand r_t in the representation E.

We also define the average joint weight $AJW(E)$ as

$$AJW(E) = \lim_{k\to\infty} \sum_{r_1=0}^{2^k} \cdots \sum_{r_d=0}^{2^k} \frac{JW_E(r_1,\ldots,r_d)}{k2^{dk}}.$$

It is obvious that $AJW(bin) = 0.75$. Solinas [21] propose the minimal weight representation for the digit set $\{0, \pm 1\}$, $d = 2$, and call the proposal as JSF. He can prove that $AJW(JSF) = 0.50$.

2.2 Average Weight on Any Expansions

In [15, 16], we discuss the method for finding the minimal average joint weight for any binary expansion. Most of the methods proposed in the literatures are focusing on finding the mathematical construction of the representation. Then, they analyze that the construction and find the minimal average weight. The advantage of this method is that they can derive the efficient conversion algorithm from the construction. But, the construction is hard to be found in some representations. For instance, there is still no work able to find the average joint hamming weight of the representation of the digit set $\{0, \pm 1, \pm 3\}$ when $d \geq 2$. Instead of finding the mathematical construction, we propose the conversion algorithm that can be applied to any digit sets. Then, we construct the analysis automatically from the conversion algorithm. Algorithm 1 shows the minimal weight conversion from $r_1,\ldots,r_d$ to our representation. We call our minimal weight representation as $MIN\{Ds,d\}$, when Ds is the desired digit set, and d is the number of scalars in the scalar point multiplication. We assume that $\max_i \log_2(r_i) = n$. We prove that this algorithm is the minimal weight conversion in [15].

From Algorithm 1, we propose Algorithm 2 to construct the Markov chain $A = (Q_A, \Sigma, \sigma_A, I_A, P_A)$, where Q_A is the set of states, Σ is the set of alphabet, σ_A is the set of transition, I_A is the initial possibility, and P_A is the transition possiblity. We consider Algorithm 1 Lines 5-20 as a function which MW outputs w. The input of the function is $bin_i(r_1,\ldots,r_d)$ and lw. We note that we do not consider lQ and Q here, as we are considering only the minimal weight not the solution. This function is referred in Line 9 of Algorithm 2.

Algorithm 1 Minimum joint weight conversion to any digit sets Ds in the binary expansion

Require: $r_1,\ldots,r_d$
The desired digit set Ds
Ensure: $MIN\{Ds,d\}(r_1,\ldots,r_d)$
1: Let Cs be a carry set such that for all $c \in Cs$ and $d \in Ds$, $\frac{c+d}{2}, \frac{c+d+1}{2} \in Cs$.
We discuss the construction of Cs in [15].
2: Let lw be an array of $lw_{c_1,\ldots,c_d}$ for any $c_1,\ldots,c_d \in Cs$.
$lw_{c_1,\ldots,c_d} \leftarrow 0$ if $\langle c_1,\ldots,c_d \rangle = \langle 0,\ldots,0 \rangle$.
$lw_{c_1,\ldots,c_d} \leftarrow \infty$ otherwise.
3: Let $lQ \leftarrow \langle lQ_{p,c_1,\ldots,c_d} \rangle$ for any $1 \leq p \leq d$ and $c_1,\ldots,c_d \in Cs$. All $lQ_{p,c_1,\ldots,c_d}$ are initiate to a null string.
4: **for** $i \leftarrow n-1$ to 0 **do**
5: **for all** $G = \langle g_i \rangle_{i=1}^d \in Cs^d$ **do**
6: $ae \leftarrow bin_i(r_1,\ldots,r_d) + G$
7: **for all** $e = \langle e_i \rangle_{i=1}^d \in Ds^d$ **do**
8: **if** $2|(ae_p - e_p)$ for all $1 \leq p \leq d$ **then**
9: $CA \leftarrow \langle \frac{ae_i - e_i}{2} \rangle_{i=1}^d$
10: $we_E \leftarrow lw_{CA}$ if $e = \langle \mathbf{0} \rangle$.
$we_E \leftarrow lw_{CA} + 1$ otherwise.
11: **else**
12: $we_E \leftarrow \infty$
13: **end if**
14: **end for**
15: Let we_{EA} is the minimal value among we.
16: $w_G \leftarrow we_{EA}$
17: Let $EA = \langle ea_i \rangle_{i=1}^d$.
18: $CE \leftarrow \langle \frac{ae_i - ea_i}{2} \rangle_{i=1}^d$
19: $Q_{s,G} \leftarrow \langle lQ_{s,CE}, ea_s \rangle$ for all $1 \leq s \leq d$
20: **end for**
21: $lw \leftarrow w$, $lQ \leftarrow Q$
22: **end for**
23: Let $Z \leftarrow \langle \mathbf{0} \rangle$.
$MIN\{Ds,d\}(r_1,\ldots,r_d) \leftarrow \langle Q_{i,Z} \rangle_{i=1}^d$

Let C be a number of states. We number each state $d \in Q_A$ as d_i where $1 \leq i \leq C$. Let $\pi^t = (\pi_i^t)$ be a probablistic distribution at time t, i.e. π_i^t is the possibility that we are on state d_i after received input length t. Next, let $P = (P_{ij})$ be the transition matrix such that

$$P_{ij} = \sum_{G \in \Sigma} P_A(i,G,j).$$

Without loss of generality, assume $d_1 = lwI$, then $\pi^0 = (1,0,...,0)^t$. From the equation $\pi^{t+1} = \pi^t P$, we find the stationary distribution such that $\pi^{t+1} = \pi^t$ by the eigendecomposition.

The next step is to find the average weight from the stationary distribution π. Define WK as a function from σ_A to the set of integer by

$$WK(lw1, G, lw2) = lw2_{\mathbf{0}} - lw1_{\mathbf{0}}$$

or the change of the hamming weight in the case that carry pair is $\langle \mathbf{0} \rangle$. We compute the average

Algorithm 2 Construct the Markov chain used for finding the average minimal weight and the minimal weight distribution

Require: The digit set Ds
The number of scalars d
Ensure: Markov chain $A = (Q_A, \Sigma, \sigma_A, I_A, P_A)$
1: $\Sigma \leftarrow \{0,1\}^d$, $Q_A \leftarrow \oslash$, $\sigma_A \leftarrow \oslash$
2: The carry set Cs is the same as the carry set in Algorithm 1.
3: $lwI \leftarrow \langle lwI_{c_1,\ldots,c_d}\rangle_{c_1,\ldots,c_d \in Cs}$, where $lwI_{0,\ldots,0} \leftarrow 0$ and $lwI_{c_1,\ldots,c_d} \leftarrow \infty$ otherwise
4: $Qu \leftarrow \{lwI\}$
5: **while** $Qu \neq \oslash$ **do**
6: let $x \in Qu$
7: $lw \leftarrow x$, $Qu \leftarrow Qu - lw$
8: **for all** $bin_i(r_1,\ldots,r_d) \in \Sigma$ **do**
9: $w \leftarrow MW(bin_i(r_1,\ldots,r_d), lw)$
10: $\sigma_A \leftarrow \sigma_A \cup \{(lw, bin_i(r_1,\ldots,r_d), w)\}$
11: $P_A(lw, bin_i(r_1,\ldots,r_d), w) \leftarrow \frac{1}{|\Sigma|}$
12: **if** $w \notin Q_A$ and $w \neq lw$ **then**
13: $Qu \leftarrow Qu \cup \{w\}$
14: **end if**
15: **end for**
16: $Q_A \leftarrow Q_A \cup \{lw\}$
17: **end while**
18: $I_A(lw) \leftarrow 1$ if $lw = lwI$, $I_A(lw) \leftarrow 0$ otherwise.

hamming weight by the average value of the change in the hamming weight when n is increased by 1 in the stationary distribution formalized as

$$AJW(MIN\{Ds,d\}) = \sum_{(x,G,y)\in\sigma_A} \frac{\pi_x WK(x,G,y)}{|\Sigma|}$$

By using this method, we can find the average joint hamming weight of many digit sets. These include

$$AJW(MIN\{\{0,\pm1,\pm3\},2\} = \frac{281}{786} \approx 0.3575,$$

$$AJW(MIN\{\{0,\pm1,\pm3,\pm5\},2\} = \frac{1496369}{4826995} \approx 0.3100.$$

3 Minimal Weight Distribution

3.1 Proof of Normal Distribution

In this section, we prove that the distribution of the joint hamming weight is the normal distribution. This generalizes a proof which shows that JSF is the normal distribution proposed by Grabner, Heuberger, and Prodinger [22]. Also, our proof generalizes the proof that the window-JSF is the normal distribution [23].

Here, we refer to the approximation theorem, Theorem 9.3 of [24].

Lemma 3.1. *Let $X_1, X_2, \ldots, X_n$ be an independent trials process and let $S_n = X_1 + X_2 + \cdots + X_n$. Assume that the greatest common divisor of the differences of all the values that the X_j can take on is 1. Let $E(X_j) = \mu$ and $V(X_j) = \sigma^2$. Then,*

$$\lim_{n\to\infty} \Pr(S_n = j) = \frac{\phi(x_j)}{\sqrt{n\sigma^2}},$$

where $x_j = \frac{j-n\mu}{\sqrt{n\sigma^2}}$, and $\phi(x)$ is the standard normal density.

Then, we prove the theorem.

Theorem 3.2. *The distribution of the joint hamming weight of $MIN\{Ds,d\}$ produced by Algorithm 1 is the normal distribution.*

Proof. Let X be a random variable which is equal to the joint hamming weight increases by one step of the Markov chain constructed in Algorithm 2. Hence, let

$$C_t = \{x \in Q_A | \sum_{G\in\Sigma, y\in Q_A} \frac{WK(x,G,y)}{|\Sigma|} = t\},$$

$$\Pr(X = t) = \sum_{x\in C_t} \pi_x.$$

This means $\Pr(X = t)$ is the possibility that the Markov chain is on the state which will increase the joint hamming weight by t in the next step. Since the function WK always returns a finite integer, the set of the possible values of t is also finite. We show the probability density function of X when $d = 2$ and $Ds = \{0, \pm1\}$ in Table 1, and when $d = 2$ and $Ds = \{0, \pm1, \pm3\}$ in Table 2.

Let X_i be the joint hamming weight increases in step i. The joint hamming weight of the bit string in the binary expansion length n, S_n, is $\sum_{i=0}^{n-1} X_i$. Then, S_n satisfies Lemma 3.1, and the distribution of S_n, the joint hamming weight, is normal. □

表 1: The probability density function of the joint hamming weight increases in one step in the Markov chain when $Ds = \{0, \pm1\}$, $d = 2$

Value	Probability
-0.25	0.25
0.25	0.125
0.5	0.125
0.75	0.25
1	0.125
1.25	0.125

表 2: The probability density function of the joint hamming weight increases in one step in the Markov chain when $Ds = \{0, \pm 1, \pm 3\}$, $d = 2$

Value	Probability
-0.75	0.0318
-0.5	0.00191
-0.25	0.130
0	0.130
0.25	0.207
0.5	0.181
0.75	0.219
1	0.0652
1.25	0.0349

3.2 Finding the Standard Deviation

In this subsection, we continue the proof from the previous subsection to find the standard deviation of the distribution.

Corollary 3.3. *Refer to the random variable X defined in the proof of Theorem 3.2, let*

$$V(X) = \sum_t t^2 \Pr(X = t) - AJW(MIN\{Ds, d\})^2.$$

The variance of the joint hamming weight distribution is $nV(X)$

Proof. Let the expectation value is μ and the standard deviation is σ. The probability density function of the normal distribution is

$$\Pr(X = d) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}).$$

And, the probability density function of the standard normal distribution is

$$\phi(x_j) = \frac{1}{\sqrt{2\pi}} \exp(\frac{-x_j^2}{2}).$$

By Lemma 3.1,

$$\lim_{n\to\infty} \Pr(S_n = j) = \frac{\phi(x_j)}{\sqrt{n\sigma^2}}.$$

Then, by $x_j = \frac{j-n\mu}{\sqrt{n\sigma^2}}$,

$$\lim_{n\to\infty} \Pr(S_n = j) = \frac{1}{\sqrt{2\pi n\sigma^2}} \exp(-\frac{(j-n\mu)^2}{2n\sigma^2}).$$

Hence, the distribution of the joint hamming weight is the normal distribution where the expectation value is $nE(X)$ and the variance is $nV(X)$. □

$V(X)$ is the constant value for the specific representation. Then, the variance of the joint hamming weight representation depends on n. Hence, the standard deviation, $\sqrt{nV(X)}$, depends on $\sqrt{n}$.

表 3: The standard deviation of the joint hamming weight distribution in some representations when $n = 160$.

Representation	Standard Deviation
$MIN\{\{0, \pm 1\}, 1\}$	$0.0493n$
$MIN\{\{0, \pm 1\}, 2\}$	$0.0400n$
$MIN\{\{0, \pm 1, \pm 3\}, 2\}$	$0.0370n$

As the bit length increases, the standard deviation becomes smaller compared to n. For example, in $MIN\{\{0, \pm 1\}, 2\}$, the standard deviation when $n = 100$ is $5.05 = 0.0505n$, and when $n = 160$ the standard deviation is $6.40 = 0.0400n$. We show the standard deviation of some representations when $n = 160$ in Table 3.

4 Application to Side Channel Attack

The result that the hamming weight distribution is the normal representation on the previous section can be used for presenting the countermeasure of the side channel attack. In this section, we present how to make the random representation which the hamming weight is fixed from that fact.

Here, we present Algorithm 3 to improve the proposals by Ha, Moon, Mamiya, and Miyaji. The algorithm is described as follows:

- We select the random number k in Line 2, and convert this number to the expansion $MIN\{Ds, d\}(k)$ at Line 3. If its weight is more than $\mu + \alpha\sigma$, we reject the random number k, and pick a new random number. As the hamming weight distribution is a normal distribution, the positive real number α determine the proportion of an integer k that can be used in our algorithm. If $\alpha = 2$, 97.73% of the set $\mathbb{Z} \cap (0, 2^n)$ can be used. If $\alpha = 2.5$, 99.38% can be used. The fact that we cannot use all k in the domain is the main drawback of our algorithm. Selecting appropriate α can trade off between the proportion of the usable scalar and the efficiency of the algorithm.
- For example, using $MIN\{\{0, \pm 1\}, 1\}(k)$ when $n = 160$, the average hamming weight is $\mu = \frac{1}{3} \times 160 \approx 53.3$, and the standard deviation is $\sigma = \sqrt{\frac{7}{18} \times 160} \approx 7.89$. If $\alpha = 2$, the fixed weight is $53.3 + 2 \times 7.89 \approx 69.1$. If $\alpha = 2.5$, the fixed weight is $53.3 + 2.5 \times 7.89 \approx 73.0$. Both numbers improve the efficiency of the existing proposal by Mamiya and Miyaji that the fixed weight is 80. Also, it improves the work by Ha and Moon that the average weight is 80.

- The function $CONVERT$ in Line 5 is defined in Algorithm 4. It is the algorithm to find the random representation with a fixed weight. We note that the function $CONVERT$ defined in Algorithm 4 is specified for $MIN\{\{0,\pm1\},1\}$ representation, but the algorithm for other representations can be referred easily.

Algorithm 3 Our proposed countermeasure of the side channel attack using the random and fixed-hamming weight representation

Require: a point on the elliptic curve P,
a bit length n,
μ is the average hamming weight of a specific representation ($\frac{1}{3}n$ when $d=1, Ds=\{0,\pm1\}$),
σ is the standard deviation of a specific representation ($\sqrt{\frac{7}{18}n}$ when $d=1, Ds=\{0,\pm1\}$),
α is a positive real number
Ensure: A random number $k \in \mathbb{Z} \cap (0, 2^n)$,
a random expansion $RAN\{Ds,d\}(k)$
a point on the elliptic curve kP.
1: **repeat**
2: Generate a random number k
3: Convert k to $MIN\{Ds,d\}(k)$
4: **until** $JW_{MIN\{Ds,d\}}(k) \leq \mu + \alpha\sigma$
5: $RAN\{Ds,d\}(k) \leftarrow CONVERT(k, \mu+\alpha\sigma)$
6: Compute kP using the double-and-add method and $RAN\{Ds,d\}(k)$.

Last, we give some explanations of Algorithm 4.

- In Line 2, we divide the bit string $MIN\{\{0,\pm1\},1\}(k)$, into m bit strings $\langle P\{i\}\rangle_{i=0}^{m-1}$. For example, $\langle 0,1,0,-1\rangle$, which is $MIN\{\{0,\pm1\},1\}(3)$, is divided into $P\{0\}=\langle 0,1\rangle$ and $P\{1\}=\langle 0,-1\rangle$.
- Since our task is to increase the hamming weight of the expansion, we need to change some zero bits of each $P\{i\}$ into 1 or -1. In the algorithm proposed by Ha and Moon, some $\langle 0,1\rangle$ or $\langle 0,-1\rangle$ is changed to $\langle 1,-1\rangle$ or $\langle -1,1\rangle$ randomly. We propose that $\langle 0,0,1\rangle$ can be changed to $\langle 1,-1,-1\rangle$, and $\langle 0,0,-1\rangle$ can be changed to $\langle -1,1,1\rangle$. Moreover,
$$\langle 0,0,\ldots,c\rangle = \langle c,-c,\ldots,-c\rangle.$$
In Lines 6-10, we changed the bit string $P\{i\}$ to the bit string $R\{i\}$ which the hamming weight is u_i+1 by the idea stated above.
- As the hamming weight of the bit string $R\{i\}$ is u_i+1, the hamming weight of the output $RAN\{\{0,\pm1\},1\}(k)$ is
$$\sum_{i=0}^{m-1} W(R\{i\}) = \sum_{i=0}^{m-1}(u_i+1) = \beta - m + m = \beta$$

Algorithm 4 The conversion to the random representation with a fixed hamming weight

Require: A key $k \in (0, 2^n)$,
an expected hamming weight β
Ensure: a fixed hamming weight random expansion $RAN\{\{0,\pm1\},1\}(k)$
1: $m \leftarrow W_{MIN\{\{0,\pm1\},1\}}(k)$
2: Let $P\{i\}$ be a bit string such that $MIN\{\{0,\pm1\},1\}(k) = \langle P\{i\}\rangle_{i=0}^{m-1}$, where $P\{i\}_j$ if $j \neq 0$, and $P\{i\}_0 \in \{-1,1\}$.
3: $V = \langle v_i\rangle_{i=0}^{m-1} \leftarrow \langle |P\{i\}|\rangle_{i=0}^{m-1}$
4: Let $U = \langle u_i\rangle_{i=0}^{m-1}$ be a random tuple of integers such that $0 \leq u_i < v_i$ and $\sum u_i = \beta - m$
5: **for** $i = 0$ to $m-1$ **do**
6: **if** $u_i = 0$ **then**
7: $R\{i\} \leftarrow P\{i\}$
8: **else**
9: Let $R\{i\} = \langle R\{i\}_j\rangle_{j=0}^{v_i-1}$ be a bit string such that
$R\{i\}_0 = \cdots = R\{i\}_{u_i-1} \leftarrow -P\{i\}_0$,
$R\{i\}_{u_i} \leftarrow P\{i\}_0$,
$R\{i\}_{u_i+1} = \cdots = R\{i\}_{v_i-1} \leftarrow 0$.
10: **end if**
11: **end for**
12: $RAN\{\{0,\pm1\},1\}(k) \leftarrow \langle R\{i\}\rangle_{i=0}^{m-1}$

5 Conclusion and Future Works

Our result proposed in this paper is the subsequence of [15, 16]. We use the Markov chain automatically generated for finding the expected value of the minimal joint hamming weight of $MIN\{Ds,d\}$ to find its distribution. As a result, we show that the distribution is the normal distribution. Then, 97.73% of the scalars used as a key has the minimal joint hamming weight less than $\mu+2\sigma$, when μ is the average joint weight, and σ is the standard deviation. We propose to reject the scalars that have more weight than that value, and propose the random and fixed-hamming-weight expansion. In $MIN\{\{0,\pm1\},1\}$, we can improve the result by Mamiya, Miyaji, Ha, and Moon from 0.50 to 0.43.

This result is still on-going because of many reasons. First, Algorithm 3 and Algorithm 4 scan the input left-to-right and right-to-left many times. Although, it is negligible if the point double and the point addition are much slower, it should be improved. Second, the random tuple generated in Line 3 of Algorithm 4 is very important for securing the scheme against the side-channel attacks. Uniform random might make the scheme weak against them. And, we need to decide the most suitable randomization method to cope with the problem. Last, we need to show that using this scheme is stronger than the other randomized expansion, in

the case that the eavesdropper iteratively uses the same scalar k to get some information.

参考文献

[1] J. C. Ha and S. J. Moon, "Randomized signed-scalar multiplication of ecc to resist power attacks," *LNCS*, vol. 2523, pp. 551–563, 2003.

[2] S. Yen, C. Chen, S. Moon, and J. Ha, "Improvement on Ha-Moon randomized exponentiation algorithm," *LNCS*, vol. 3506, pp. 154–167, 2005.

[3] H. Mamiya and A. Miyaji, "Fixed-hamming-weight representation for indistinguishable addition formulae," *IPSJ Journal*, vol. 47, pp. 2430–2439, August 2006.

[4] M. Joye, "Elliptic curves and side-channel attacks," *ST Journal on System Research*, vol. 4, pp. 17–21, September 2003.

[5] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," *LNCS*, vol. 1109, pp. 104–113, 1996.

[6] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *LNCS*, vol. 1666, pp. 388–397, 1999.

[7] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," *LNCS*, vol. 2523, pp. 29–45, 2002.

[8] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystem," *LNCS*, vol. 1717, pp. 292–302, 1999.

[9] E. Brier and M. Joye, "Weierstrab elliptic curves and side-channel attacks," *LNCS*, vol. 2274, pp. 335–345, 2002.

[10] B. Chevallier-Mames, M. Ciet, and M. Joye, "Low-cost solutions for preventing simple side-channel analysis: Side channel atomicity," *IEEE Trans. on Comp.*, vol. 53, pp. 760–768, 2004.

[11] D. Boneh, R. A. DeMilo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," *LNCS*, vol. 1233, pp. 37–51, 1997.

[12] P. Fouque, F. Muller, G. Poupard, and F. Valette, "Defeating countermeasures based on randomized BSD representations," *LNCS*, vol. 3156, pp. 312–327, 2004.

[13] K. Okeya and D. Han, "Side channel attack on Ha-Moon's countermeasure of randomized signed scalar multiplication," *LNCS*, vol. 2904, pp. 159–187, 2004.

[14] J. Shin, D. Park, and P. Lee, "DPA attack on the improved Ha-Moon algorithm," *LNCS*, vol. 3786, pp. 283–291, 2006.

[15] V. Suppakitpaisarn, "Optimal average joint hamming weight and digit set expansion on integer pairs," Master's thesis, The University of Tokyo, 2009.

[16] V. Suppakitpaisarn and M. Edahiro, "Fast scalar-point multiplication using enlarged digit set on integer pairs," *Proc. of SCIS 2009*, p. 14, 2009.

[17] J. Jedwab and C. Mitchell, "Minimum weight modified signed-digit representations and fast exponentiation," *Electronics Letters*, vol. 25, no. 17, pp. 1171–1172, 1989.

[18] O. Egecioglu and C. K. Koc, "Exponentiation using canonical recoding," *Theoretical Computer Science*, vol. 129, pp. 407–417, 1994.

[19] J. A. Muir and D. R. Stinson, "New minimal weight representation for left-to-right window methods," *Department of Combinatorics and Optimization, School of Computer Science, University of Waterloo*, 2004.

[20] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, pp. 36–63, August 2001.

[21] J. A. Solinas, "Low-weight binary representation for pairs of integers," *Centre for Applied Cryptographic Research, University of Waterloo, Combinatorics and Optimization Research Report CORR*, 2001.

[22] P. J. Grabner, C.Heuberger, and H. Prodinger, "Distribution results for low-weight binary representations for pairs of integers," *Theoretical Computer Science*, vol. 319, pp. 307–329, 2004.

[23] P. J. Grabner, C. Heuberger, H. Prodinger, and J. M. Thuswaldner, "Analysis of linear combination algorithms in cryptography," *ACM Transactions on Algorithms*, vol. 1, pp. 123–142, July 2005.

[24] C. M. Grinstead and J. L. Snell, *Introduction to probability.* The American Mathematical Society, 2nd. ed., July 2006.